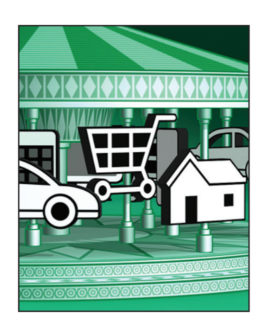


From the Service-Oriented Architecture to the Web API Economy

Wei Tan • *IBM T.J. Watson Research Center*
Yushun Fan • *Tsinghua University, China*
Ahmed Ghoneim and M. Anwar Hossain • *King Saud University, Saudi Arabia*
Schahram Dustdar • *TU Wien, Austria*

As Web APIs become the backbone of Web, cloud, mobile, and machine learning applications, the services computing community will need to expand and embrace opportunities and challenges from these domains.

Service-oriented architecture (SOA) made its debut in the early 2000s as a new architecture pattern. In this pattern, software components are encapsulated as individual services (or Web APIs), and invoked from the network through standard Web protocols such as HTTP. Web APIs are lightweight alternatives to WSDL/SOAP-based services that usually use REST as the communication protocol and JSON as the content format.[1] A service usually represents a minimal reusable component that can be combined with many other services, forming value-added business processes, also known as composite services. The SOA paradigm with the accompanying Web service protocols — including SOAP, REST, Web Service Definition Language (WSDL), and Web Services Business Process Execution Language (WS-BPEL) — has become the de facto standard in enterprise information systems to achieve interoperability.[1]

These days, SOA and services computing have gone much beyond interoperation technology (see Figure 1). REST-style Web APIs have replaced SOAP services for two reasons: first, REST's create, read, update, and delete (CRUD) interface greatly improves consumability; second, JSON with REST makes the communication payload much simpler and easier to understand, compared to XML with SOAP. As evidence, starting in 2006 Google abandoned SOAP and only uses REST in its search APIs. While SOAP/WSDL is still popular in many enterprise systems, REST-style Web APIs are pervasive in Web, mobile, cloud infrastructure, and Internet of Things (IoT) applications. According to ProgrammableWeb (http://programmableWeb.com), the largest online API registry, Web API enjoyed a compounded annual growth rate of 100 percent (approximately) from 2005 to 2011, in terms of the total number of APIs registered. As of March 2016, ProgrammableWeb has listed more than 14,700 APIs. With the formation of this API ecosystem, an API economy is emerging. To give you a few examples, 60 percent of Salesforce's transactions go through its APIs instead of the traditional Web GUI, contributing to its 1.3 billion daily transactions and more than $5 billion in annual revenue. Additionally, 90 percent of Expedia, 60 percent of eBay, and 100 percent of Amazon Web Services (AWS) revenue are from APIs.[2]

Emerging Application Domains of Web APIs

Web services, particularly Web APIs, are becoming the backbone of Web, cloud, mobile, and machine learning applications. Thus, we argue that the services computing community should extend its scope, and embrace the newly emerged opportunities and challenges from these domains.

Web Applications

Web application developers can create a service composition (that is, a *mashup*), by combining multiple services. For example, we can create a trip itinerary service by combining a map service with a flight, a train, a rental car, and a hotel

 1089-7801/16/$33.00 © 2016 IEEE

booking service. ProgrammableWeb has listed more than 6,000 mashups (www.programmableWeb.com/category/all/mashups).

Cloud Applications

Infrastructure-as-a-service cloud services, such as computing services that provide virtual machines, storage services that provide block or object storage, and message services that provide reliable message queues, are becoming the "utility providers" of many Internet businesses. As an example, Netflix uses various services from AWS, including virtual machines, storage, message queues, and databases.[3] As another example, Dropbox (which offers online file storage and synchronization services) stores all its customer files on Amazon Simple Storage Service (S3).

Mobile and IoT Services

Nowadays, the so-called IoT — including smartphones, vehicles, wearables, smart home appliances, and smart factory machines — are connected to the cloud and the Web, or interconnected with one another. Studies have shown that many Web APIs such as advertising, social network, messaging, and billing[4,5] are commonly used in mobile apps (see Figure 2).

Machine Learning and Big Data Services

Machine learning services are becoming important in the new wave of AI hype. Business owners want to focus on their core competence and outsource some non-essential but still very important features to a third party who possesses a given expertise. This isn't a new phenomenon, but it's becoming particularly interesting in the Big Data era. First, many companies (such as electronic commerce and content streaming) need machine learning capabilities, including image recognition, natural language processing, and recommendation as an integral part of their business. However, they usually don't own the sophisticated machine learning algorithm, the sufficient training data to deliver a decent model, or a computer system capable of handling a big volume of training data to derive a model in a timely manner. This is a sweet spot for services computing — occupying the niche where a business wants to use a certain capability, but others are in a better position to deliver that.

Currently, two categories exist for machine learning services. The first category is a platform service through which users provide their own training data and receive a trained model. After training with customer data (training can take much time, depending on the data size and model complexity), the derived model can either be retrieved or stay in the same cloud to serve the inference purpose. Services of this category include Amazon Machine Learning (https://aws.amazon.com/machine-learning), Microsoft Azure Machine Learning (https://azure.microsoft.com/en-us/services/machine-learning) and Google Prediction API (https://cloud.google.com/prediction). An exemplary use case of that is to upload many emails marked as spam or non-spam, to train a classification model that acts as a spam filter for future emails.

The second category is a software service where users don't provide any training data. Instead, users only provide the data to be inferenced. Services of this category include Google Translate API (https://cloud.google.com/translate) and IBM Watson Developer Cloud (www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud). An exemplary use case of that is to upload a sentence in Chinese and get its translation in English, or to upload a photo and let the API recognize the objects in it.

In the following, we discuss two case studies that further explore the machine learning and IoT categories.

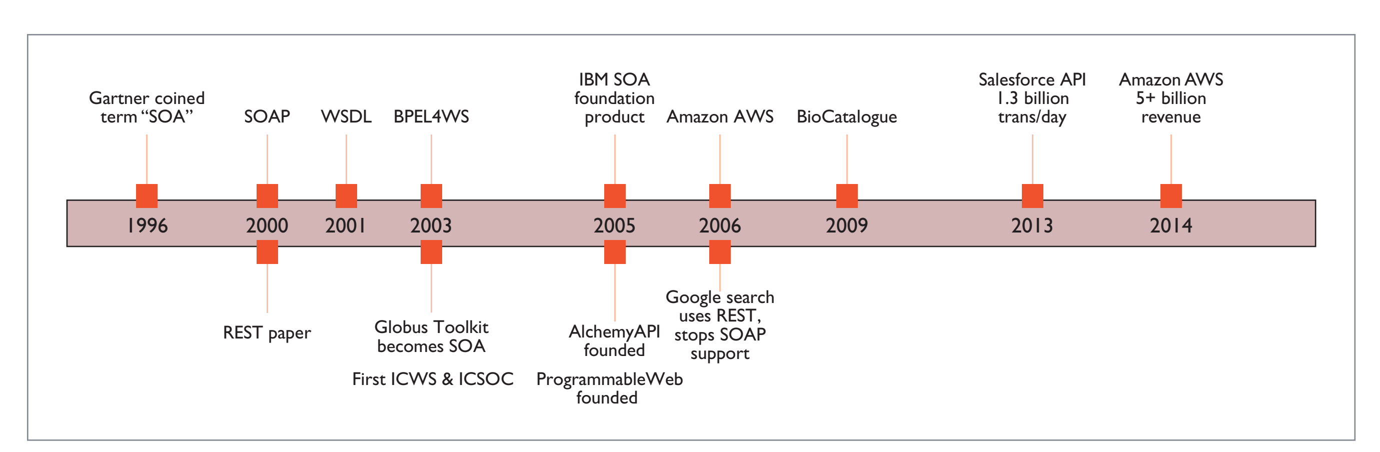


Figure 1. A brief history of services computing. After debuting in the early 2000s, services computing has transitioned from an enterprise interoperation technology to shape the Web API economy. (AWS = Amazon Web Services; BPEL4WS = Business Process Execution Language for Web Services; SOA = service-oriented architecture; WSDL = Web Services Description Language.)

Case study 1. In this scenario, the focus is on a faster and cheaper recommendation service using GPU acceleration. Recommendation is a key technology for online merchant and content-streaming companies. As an example, 80 percent of Netflix's watching hours are influenced by its recommender system.[6] However, many of the recommender systems either require a sizable infrastructure or perform slowly when the number of users and items grows.[7] For instance, Chao Liu and his colleagues[8] mentioned starting their recommendation model training on Friday afternoon and getting the result by Monday morning. It follows, then, that an e-commerce website similar to Amazon or a digital content streaming provider similar to Netflix could only update the recommendation model once a week. This workaround isn't viable when business becomes global and available every day around the clock.

Our work[7] proposes using GPUs to accelerate (collaborative filtering-based) recommendation. By exploiting the massive parallelism inside individual GPU devices and across multiple devices, a matrix factorization tool called cuMF (https://github.com/wei-tan/CuMF) is able to offer recommendation services up to 10 times as fast, and up to 100 times as cost-efficient, compared to state-of-art distributed CPU solutions. This work sheds light on how a service's quality can greatly improve by adopting an advanced computing infrastructure.

Figure 2. Web, cloud, mobile/Internet of Things (IoT), and machine learning applications are fully reliant on Web services/APIs. (a) Web (mashup); (b) cloud (infrastructure); (c) mobile and IoT; and (d) machine learning

Table 1. Topics of interest to the services computing community, and their relation to other computer science areas.

Topics	Related to computer science areas
Verification, composition	Software engineering
Semantics, security, and invocation framework	Web
Workflow, provenance, and data management	Databases
Distributed services, reliability, scalability, and performance	Systems
Automatic composition, recommendation	AI
Service interface, usability	Human-computer interface (HCI)
Optimization of service composition, quality	Operational research (OR)

Case study 2. This scenario focuses on connecting people, mobile phones, and smart IoT devices. With more Internet-accessible IoT devices, connecting them via Web services with mobile phones and people is often the best way to utilize them. For example, someone might prefer to get an emergency call if his smart smoke detector raises an alarm; or perhaps that same person would like to receive a notification from a weather service if it predicts rain tomorrow, along with a reminder to take an umbrella and tell his smart irrigation device to delay watering the lawn. Mobile apps such as IFTTT (an abbreviation of "If This Then That;" see https://ifttt.com) let end users create chain APIs called "recipes," which can connect Web services such as email, social network, online photo storage, and IoT devices (including smart home appliances, vehicles, and factory machines).

The Research Community's Shifting Interest

Table 1 shows the major topics of interest to the services computing

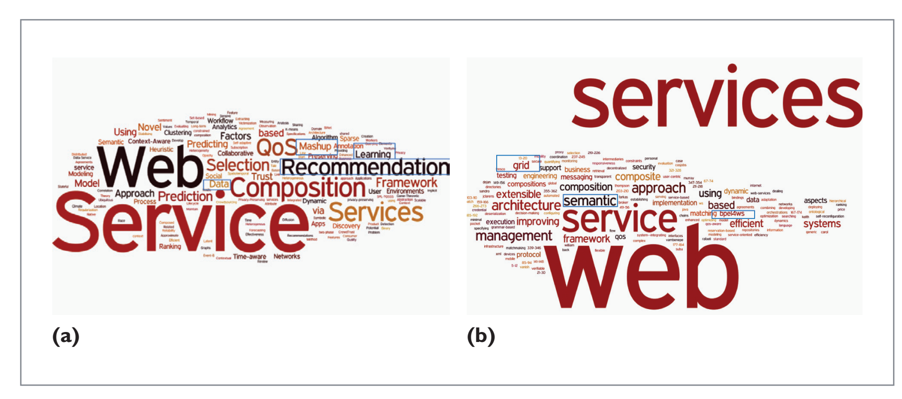


Figure 3. Word clouds of paper titles from the IEEE International Conference on Web Services for (a) 2015 and (b) 2005, respectively.

community, and their relation to traditional computer science areas. Clearly, services computing is an interdisciplinary area. It borrows methodologies and technologies from areas such as software engineering, the Web, databases, systems, AI, HCI, and operational research (OR) to tackle its specific problems.

To further gauge the community's interests over time, we collected titles from IEEE International Conference on Web Services (ICWS) papers from 2015 and 2005, respectively. The word clouds in Figure 3 clearly illustrate a focus on data, mashup, and recommendation for 2015; and a focus on semantic, grid, and BPEL4WS for 2005.

In looking at the services computing community's shift in interest over time, and determining the best paths moving forward, we offer two suggestions to researchers and practitioners:

1) Pay more attention to more practical approaches to solve real-life problems.
2) Pay more attention to applications in emerging areas such as mobile, IoT, and machine learning.

Regarding our first suggestion — let's keep in mind that in the adolescent age of services computing, studies in this area cover some ivory-tower topics such as automated composition, verification of service processes using formal methods, and Semantic Web services. Researchers came up with sophisticated formalism and methods to tackle these problems, only to find that they aren't close enough to reality. As an example, our earlier empirical study[9] on scientific workflows shows that most service compositions use only a handful of services. As a result, the service processes are relatively simple. Therefore, in most real-life cases it would be an overkill to use formal methods such as Petri nets and process algebra to analyze them. At the same time, people have found that fully automated service composition is also far from reality. A more realistic approach is to provide context-aware recommendations in a composition's design phase.[1,10–12] For example, a recent study[13] takes a highly innovative approach to facilitate developers and even end users accomplishing the composition in an interactive fashion.

Thus, when services computing moves from academia to industry shaping an API economy, people expect the technologies to be more practical and address the pain-point of developers.

Regarding our second suggestion, we argue that services or APIs shouldn't limit themselves to enterprise integration. Instead, researchers and practitioners should pay attention to emerging areas, including mobile/IoT, network,[14] Big Data,[15] and machine learning services. Because these areas are quickly developing and evolving, and have a big demand for using service as the delivery channel, we believe that services computing researchers would find this area of development more valuable.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61174169). We also extend our sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this International Research Group (IRG14-28).

References

1. W. Tan and M.C. Zhou, *Business and Scientific Workflows: A Web Service-Oriented Approach*, John Wiley & Sons, 2013.
2. M. Vukovic et al., "Riding and Thriving on the API Hype Cycle," *Comm. ACM*, vol. 59, no. 3, 2016, pp. 35–37.
3. Netflix, "Four Reasons We Choose Amazon's Cloud as Our Computing Platform," *Netflix Tech Blog*, 14 Dec. 2010; http://techblog.netflix.com/2010/12/four-reasons-we-choose-amazons-cloud-as.html.
4. A. Gorla et al., "Checking App Behavior against App Descriptions," *Proc. 36th Int'l Conf. Software Eng.*, 2014, pp. 1025–1035.
5. N. Viennot, E. Garcia, and J. Nieh, "A Measurement Study of Google Play," *The 2014 ACM Int'l Conf. Measurement and Modeling of Computer Systems*, ACM, 2014, pp. 221–233.
6. C.A. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Trans. Management Information Systems*, vol. 6, no. 4, 2015, article no. 13.
7. W. Tan, L. Cao, and L. Fong, "Faster and Cheaper: Parallelizing Large-Scale Matrix Factorization on GPUs," *ACM Proc. 25th Int'l Symp. High-Performance Parallel and Distributed Computing*, 2016, to be published.
8. C. Liu et al., "Distributed Nonnegative Matrix Factorization for Web-Scale Dyadic Data Analysis on Mapreduce," *Proc. 19th Int'l Conf. World Wide Web*, 2010, pp. 681–690.

9. W. Tan, J. Zhang, and I.T. Foster, "Network Analysis of Scientific Workflows: A Gateway to Reuse, *Computer*, vol. 43, no. 9, 2010, pp. 54–61.
10. Y. Zhong et al., "Time-Aware Service Recommendation for Mashup Creation," *IEEE Trans. Services Computing*, vol. 8, no. 3, 2015, pp. 356–368.
11. B. Xia et al., "Category-Aware API Clustering and Distributed Recommendation for Automatic Mashup Creation," *IEEE Trans. Services Computing*, vol. 8, no. 5, 2015, pp. 674–687.
12. S. Wang et al., "Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems," *IEEE Trans. Services Computing*, vol. 8, no. 5, 2015, pp. 755–767.
13. X. Liu et al., "Data-Driven Composition for Service-Oriented Situational Web Applications," *IEEE Trans. Services Computing*, vol. 8, no. 1, 2015, pp. 2–16.
14. X. Qiao et al., "Service Provisioning in Content-Centric Networking: Challenges, Opportunities, and Promising Directions," *IEEE Internet Computing*, vol. 20, no. 2, 2016, pp. 26–33.
15. Z. Zhou et al., "IEEE Access Special Session Editorial: Big Data Services and Computational Intelligence for Industrial Systems," *IEEE Access*, vol. 3, 2015, pp. 3085–3088.

Wei Tan is a research staff member at the IBM T.J. Watson Research Center. His research interests include GPU acceleration, large-scale machine learning, service-oriented architecture, business and scientific workflows, and Petri nets. Tan has a PhD in automation engineering from Tsinghua University, China. Contact him at wtan@us.ibm.com.

Yushun Fan is a professor at Tsinghua University, China. His research interests include computer integrated manufacturing, service computing, intelligent service platforms, workflow, and Big Data. Fan has a PhD in automation engineering from Tsinghua University. Contact him at fanyus@tsinghua.edu.cn.

Ahmed Ghoneim is an assistant professor in the Department of Software Engineering, College of Computer Science and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His research interests address software evolution; service-oriented engineering, software development methodologies, quality of service, net-centric computing, and human-computer interaction (HCI). Ghoneim has a PhD in software engineering from the University of Magdeburg, Germany. Contact him at ghoneim@ksu.edu.sa.

M. Anwar Hossain is an associate professor in the Department of Software Engineering, College of Computer Science and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His research interests include multimedia surveillance and privacy, ambient intelligence, the Internet of Things, and cloud computing. Hossain has a PhD in electrical and computer engineering from the University of Ottawa, Canada. He's a member of IEEE and ACM. Contact him at mahossain@ksu.edu.sa.

Schahram Dustdar is a full professor of computer science and he heads the Distributed Systems Group at TU Wien, Austria. His work focuses on distributed systems. Dustdar is an IEEE Fellow, a member of the Academia Europaea, an ACM Distinguished Scientist, and recipient of the IBM Faculty Award. Contact him at dustdar@dsg.tuwien.ac.at; dsg.tuwien.ac.at.